

REMARKS

Claims 1-33 are pending in the instant application. The Examiner rejected claims 1-11 and 14-33, and objected to Claims 12 and 13. Applicants respectfully traverse the Examiner's rejections. The Examiner also objected to the drawings have been drawn by hand. Applicants have attached formal drawings based on those originally filed. Applicants have taken care not to introduce new matter during the preparation of the formal drawings.

Claims 26-33 were rejected under 35 USC 102(e) as being anticipated by U.S. Patent No. 6,026,235 to Shaughnessy. The Shaughnessy reference discloses a development system having a monitor/profiler tool for monitoring functions in natively compiled software programs, wherein the tool is constructed to work directly on a natively compiled software application which only has debugging information, and does not require a special compile or link phase for the application under examination [Abstract]. The tool is designed to improve the machine-level code generated for a program by analyzing the code, including looking at program performance for detecting any significant performance bottlenecks, detecting invalid API usage and memory leaks, as well as performing working set and coverage analysis [Column 2, lines 36-43]. According to Shaughnessy, the tool can monitor any function in software applications which have debug information [Abstract; Column 3, lines 64-67]. Shaughnessy teaches that such debug information is required for the tool to operate [Column 3, lines 59-62; Column 4, lines 3-48; Column 8, lines 56-62], and that such debug information is typically present in executables under development because developers need to debug their applications [Column 9, lines 1-3]. Shaughnessy teaches that the functions in the application can then be found by scanning the import tables of shared libraries and debug information for a given executable file [Column 4, lines 18-27]. Shaughnessy then teaches generating patch stubs and modifying an executable or library itself, as stored on disc [Column 13, line 27-29], to facilitate function call monitoring. Such modifications include the generation of patch code and insertion of such patch code into the executable or library [Column 11, line 31 to Column 14, line 40]. The only aspect of the Shaughnessy application that takes place in memory is the creation and searching of a symbol table, which facilitates identifying function calls [Column 14, lines 38-40].

Unlike the Shaughnessy reference, the executable itself is not modified as part of Applicants' claimed invention. Alteration of the executable file itself is undesirable as it will typically alter the checksum value associated with the executable, which can prevent the operating system from loading the executable. Alteration of the executable file itself is also undesirable because such alterations will typically change the size of the file or other attributes thereof, which can result in anti-virus programs

rejecting the executable file. By contrast, in Applicants' invention, only the instance of the software that is in memory is altered, and even then, only the import table portion of the software is actually altered. Although Applicants believe Claim 26 was clear that the import table alteration took place in memory, Applicants have amended Claim 26 to further reinforce the fact that the application is loaded into memory before any alterations take place. Shaughnessy neither teaches nor suggests altering a memory-resident import table.

Furthermore, Applicants' Claim 26 recites obtaining the address of an executable program from an operating system. The Examiner cites column 9, lines 11-19, as teaching this step of Claim 26. However, Column 9, lines 11-19, simply recites that an import table exists as part of the application, and that the import table and debug information can be used to identify functions called by the application. Column 9, lines 11-19 do not teach or suggest obtaining the address of an executable program from an operating system. The Court of Appeals for the Federal Circuit has consistently held that "Anticipation requires the presence in a single prior art reference disclosure of each and every element of the claimed invention, arranged as in the claim." Lindemann Maschinenfabrik GmbH v. American Hoist & Derrick, 221 USPQ 481, 485 (Fed. Cir. 1984). Shaughnessy clearly fails to teach or suggest structure positively recited and claimed in Applicants' independent Claim 26, thus, Applicants' Claim 26 is patentable over Shaughnessy. Applicants respectfully request that the Examiner withdraw his rejection of Claim 26.

Claims 27 through 33 depend from Claim 26, and are therefore patentable for at least the reasons set forth above with respect to Claim 26. The Court of Appeals for the Federal Circuit has consistently held that where a claim is dependent upon a patentable independent claim, the independent claim is *a fortiori* patentable because it contains all the limitations of the independent claim plus further limitations. See, e.g., Hartness Intern. Inc. v. Simplimatic Engineering Co., 819 F.2d 1100, 1108 (Fed. Cir. 1987). Applicants reassert the arguments set forth above with respect to Claim 26 for each of Claims 27 through 33, and respectfully request that the Examiner withdraw his rejection of these claims.

The Examiner rejected Claims 1, 2, 5, and 25 as being obvious under 35 USC §103(a) over U.S. Patent No. 6,185,681, to Zizzi, in view of U.S. Patent No. 6,611,878 to De Armas et al. The Examiner's combination of the Zizzi and De Armas patents is believed to be motivated by hindsight, rather than by a teaching or suggestion within the prior art. As stated by the Court of Appeals for the Federal Circuit, "Determination of obviousness cannot be based on the hindsight combination of components selectively culled from the prior art to fit the parameters of the patented invention." ATD

Corp. v. Lydall, Inc., 159 F.3d 534, 546 (Fed. Cir. 1998). There must be a teaching or suggestion within the prior art, within the nature of the problem to be solved, or within the general knowledge of a person of ordinary skill in the field of the invention, to look to particular sources, to select particular elements, and to combine them as combined by the inventor. See, Ruiz v. A.B. Chance Co., 234 F.3d 654, 665, (Fed. Cir. 2000); ATD Corp., 159 F.3d at 546; Heidelberger Druckmaschinen AG v. Hantscho Commercial Prods., Inc., 21 F.3d 1068, 1072 (Fed. Cir. 1994) ('When the patented invention is made by combining known components to achieve a new system, the prior art must provide a suggestion or motivation to make such a combination.')." Crown Operations Intern., Ltd. v. Solutia Inc., 289 F.3d 1367, 1376 (Fed. Cir. 2002). The Examiner has failed to provide any teaching or suggesting within the Zizzi reference to combine it with the De Armas reference, and Applicants therefore assert that Claims 1, 2, 5, and 25 are therefore patentable over the cited references.

In addition, even if the Zizzi and De Armas references were combined, Applicants' invention, as claimed in Claims 1, 2, 5, and 25, would not be obvious over the combination. As the Examiner points out on page 6 of the Office Action, the Zizzi reference teaches trapping event messages related to opening and closing documents. As described in the Zizzi reference, event messages are generated by interaction of a user input device, such as a mouse or keyboard, with the application in a display window [Figure 4; Figure 5; Column 4, lines 27-35; Column 8, lines 32-41; Column 9, lines 43-53]. This is reinforced in the De Armas reference [Column 6, line 50, through Column 7, line 4]. By contrast, Applicants' invention, as claimed in Claim 1, is directed to associating application-based file activity with the graphical display of a file on a screen, wherein calls to operating system DLLs by the application are intercepted by fixing the import table of the application. The calls intercepted by Applicants' invention occur at a lower level than the event trapping described in the Zizzi reference, and are therefore more likely to intercept both user-initiated hard drive accesses and application-initiated hard drive accesses. Applicants' invention also includes substituting a message monitoring program window function with the application's main window function. The message monitoring program window function is designed to provide enhanced or alternative versions of the notification windows typically associated with the application's main window function. In this manner, Applicants' invention permits the application to call the enhanced notification windows associated with the message monitoring program without having to modify the application. From the user's perspective, this results in an association of file activity with the document displayed on the screen. As an example of the advantage of Applicants' invention over the combination of the Zizzi and De Armas references, the combined references would allow temporary files created by an application as

the application is in use, such as those created by Microsoft Word and other programs, to remain unencrypted. Such temporary files frequently remain on user computers well beyond their use by the application, and represent a significant security loophole. By intercepting hard drive accesses at a lower level and substituting a monitoring program window function for the applications' main window function, Applicants invention is able to provide stronger encryption than that afforded by the combination of the Zizzi and De Armas references. Furthermore, by associating file activity with the graphical display of a file on the screen, Applicants' invention can take action on the displayed file, and all associated files, at the time the file is closed by or for the user.

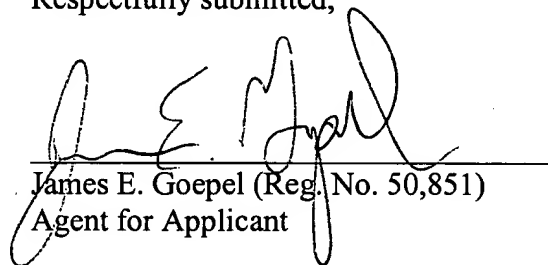
Clearly, neither Zizzi nor De Armas teach or suggest fixing the import table of the application with addresses of functions from a message monitoring program library and substituting the applications' main window function with a message monitoring program window function to associate application file activity with the graphical display of a file on a screen. It is well established that, in order to show obviousness, all limitations must be taught or suggested by the prior art. In Re Boyka, 180 U.S.P.Q. 580, 490 F.2d 981 (CCPA 1974); MPEP § 2143.03. It is error to ignore specific limitations distinguishing over the references. In Re Boe, 184 U.S.P.Q. 38, 505 F.2d 1297 (CCPA 1974); In Re Saether, 181 U.S.P.Q. 36, 492 F.2d 849 (CCPA 1974); In Re Glass, 176 U.S.P.Q. 489, 472 F.2d 1388 (CCPA 1973). The combination of Zizzi and De Armas does not teach or suggest all limitations of Applicants' claimed invention as set forth in Claim 1, and Claim 1 is therefore patentable for at least this reason. Applicants respectfully request that the Examiner withdraw his rejection of Claim 1.

The combination of Zizzi and De Armas also forms the basis for the Examiner's rejection of Claims 2 through 25. Applicants therefore assert that Claims 2 through 25 are patentable for at least the reasons set forth above with respect to Claim 1. Applicants therefore respectfully request that the Examiner withdraw his rejection of Claims 2 through 25.

CONCLUSION

Having responded to all objections and rejections set forth in the outstanding Office Action, it is submitted that claims 1 through 33 are in condition for allowance and Notice to that effect is respectfully solicited. In the event that the Examiner is of the opinion that a brief telephone or personal interview will facilitate allowance of one or more of the above claims, he is courteously requested to contact applicant's undersigned representative.

Respectfully submitted,



James E. Goepel (Reg. No. 50,851)
Agent for Applicant

GREENBERG TRAURIG, LLP
1750 Tyson's Boulevard, Suite 1200
McLean, VA 22102
(703) 903-7536
E-mail: goepelj@gtlaw.com

Filed: January 20, 2004